The Fifth International Conference on Neural Information Processing
Kitakyushu, Japan October 21-23,1998

Active Data Partitioning for Building Mixture Models

Suk-Joon Kim and *Byoung-Tak Zhang*
Email: {*sjkim, btzhang*}*@scai.snu.ac.kr*

Artificial Intelligence Lab (SCAI)
Dept. of Computer Engineering, Seoul National University
Seoul, 151-742, Korea

ABSTRACT

This paper introduces two data partitioning methods for building mixtures of several neural networks. The methods are based on active learning with two different selection measures. One is the redundant data selection (RDS) method which chooses examples with less error, and the other is the critical data selection (CDS) method which chooses examples with larger error. The partitioned data sets are used to train the experts which are then combined by a weighted majority algorithm to produce final outputs. Experiments have been performed on two data sets from the UCI machine learning database. The results show that CDS outperforms both RDS and random selection in generalization ability. We also suggest a promising way to use the data subsets partitioned by RDS.

KEYWORDS: data partitioning, active learning, mixture of experts

1. Introduction

It has been shown that if a problem domain can be divided into several subtasks, the overall performance can be enhanced by an effective combination of expert neural networks. Hampshire [1] described a system of this kind that can be used when the division into subtasks is known prior to training. Jacobs [3] have described a related system that learns how to allocate cases to experts. Igor [2] and Shadafan [6] suggest data partitioning algorithms to separate the entire training data into several subsets by its characteristics. In [2], a correlation coefficient matrix per every input data should be maintained and 200 expert networks are required for the partitioning. In [6], three types of matrix operations per every input data and additional large data structures per node are required for the partitioning. Although these algorithms are good at partitioning the training data set into subsets with different characteristics, they require large computational cost.

In this paper we present new data partitioning methods. The methods are based on the active learning paradigm in which the learner actively selects new training examples incrementally during learning [8, 9]. It has been shown that the active learning can estimate the global distribution using a small portion of the whole data set [8]. The basic idea in the present paper is that the active data selection algorithm can also be used to partition the entire data into sebsets for effective construction of mixture models. The method is applied to the problems in the UCI machine learning database. Our experiments show that this kind of mixtures can enhance the performance if the predictions of each experts are properly combined.

The organization of this paper is as follows. Section 2 presents new data partitioning methods. Section 3 describes the mixture model to build an ensemble network using the partitioned subsets. Section 4 reports experimental results. Section 5 draws conclusions and discusses future work.

2. Active Partitioning of Training Data

Conventional neural network algorithms assume that the training data are given from the environment or an external oracle. Thus, the learning is focused on the adjustment of the learner's free parameters. On the other hand, several researchers have studied the active learning paradigm in which the learner selects training examples actively from its environment [5, 4, 8, 9].

Active learning can be used to find a subset that is as small as possible and at the same time contains as much information as possible. Zhang [8] defines the example causing maximum error for the current trained network as the most critical. That is, the criticality is proportional to the mean error and computed by the current trained neural network (W, A):

$$e_m(s) = \frac{1}{\dim(y_m)} \|y_m - f(x_m; W, A)\|,$$

where (x_m, y_m) is the m-th training pattern, f is the output of the network with weights W and architecture A and s is the number of selection. An example with a maximum error e_m is selected from the candidate set C_s:

$$m^* = \underset{m \in C}{\operatorname{argmax}}(e_m(s)).$$

This scheme tends to sequentially select examples that represent the global distribution of the candidate set. In this paper, we call the above selection scheme as *critical data selection* (CDS). The data partitioning occurs when the selected data is sequentially accumulated for a fixed period. The resulting subsets approximate the global distribution of the whole data.

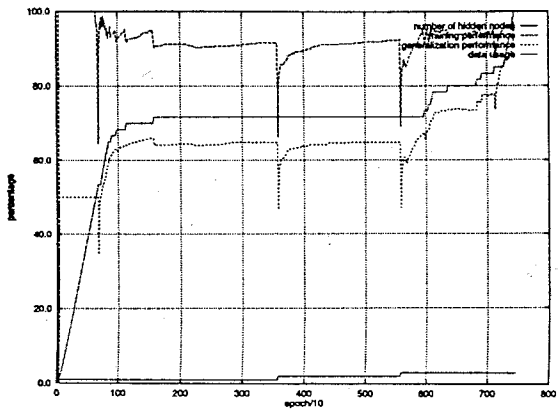
Another possibility of data partitioning using active learning is to select examples with minimum errors, rather than maximum errors:

$$m^* = \underset{m \in C}{\operatorname{argmin}}(e_m(s))$$

This method prefers to select examples that are similar to the already selected examples, thus representing local regions of the example space. This kind of selection scheme will be referred to as *redundant data selection* (RDS). The active data partitioning algorithm can be described as follows.

Step 0. Initialize the candidate set C, class number k, selection step i, and newly selected data set D_{new}:
$C \leftarrow$ initial candidate set,
$k \leftarrow 1$, $i \leftarrow 1$, $D_{new} \leftarrow \emptyset$.

Step 1. Initialize the training set D and network architecture A:



re 1: Partitioning points during the active learning. First th of the network at the 3550th epoch is the starting t for the new partitioning.

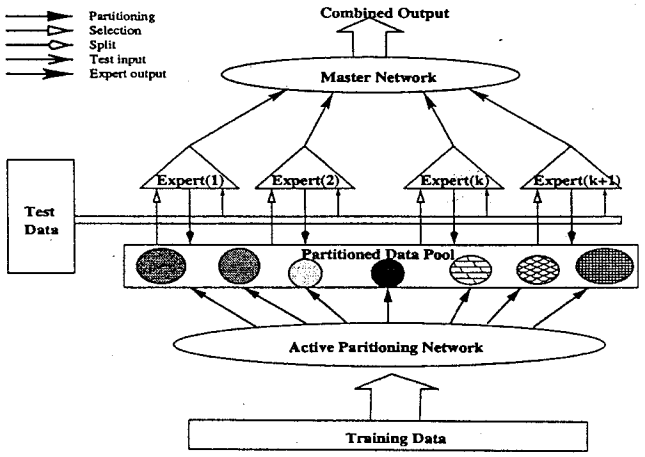


Figure 2: Schematic diagram of the committee system. The active network partitions the training data set into several subsets. Each expert is trained with a random pair of the sets in the partitioned data pool. Given an input, each expert makes a decision and their results are combined by the master network to produce the final output.

$D \leftarrow$ one example of class k,
$A \leftarrow$ single hidden node.

ep 2. Train the network A for maximum number of epochs or until $E_t(W; A, D) < E_{min}$, where $E_t(W; A, D)$ is the training error and E_{min} is the desired training error level.

ep 3. If $E_t(W; A, D) \geq E_{min}$, then go to step 5.

ep 4. If $|C|$ =0, goto step 6,
else select examples m^* from C and
$D \leftarrow D \cup (x_{m^*}, y_{m^*})$, $D_{new} \leftarrow D_{new} \cup (x_{m^*}, y_{m^*})$,
then got step 2.

p 5. Grow the network A and set the i-th new partition $S_i \leftarrow D_{new}$.
$D_{new} \leftarrow \emptyset$, $i \leftarrow i + 1$.

p 6. If $k \geq N_{class}$, then stop, else $k \leftarrow k + 1$ and goto step 1.

e 1 shows the learning curve of the described algorithm. urve shows that the architecture grows at the 3550th , and step 5 in the described algorithm is performed at ne. That is, the selected data D_{new} is stored as a new ned subset, and then another new partitioning session at this point. As shown in Figure 3 in the next page, s 2 data set is designed to have three clusters, but the partition algorithm partitioned it into two partitions, is more acceptable when considering the distribution 1.

Building a Committee Using Partitioned Data

hematic diagram of the overall committee system is in Figure 2. The data set is partitioned into sub- ough the active network which are stored in the data the partitioning is performed by CDS, each expert is with a randomly selected subset from the data pool. al prediction is computed by combining all the pre- of experts.
S is used for partitioning, the partitioned data sets in al are randomly selected by each expert and then the training begins. On training, the selected data set is partitioned again into two subsets if E_{min} is not reached after sufficient training. This *subpartitioning* process is useful to make each expert as simple as possible. Since the trained expert learns only a portion of the entire space, this expert is a kind of *local expert* [3].
There are several candidates for the master algorithm. This paper considers the weighted majority (WM) algorithm only, which is widely used in the mixture models. The weighted majority algorithm makes predictions by taking a weighted vote among the experts and learns by altering the weights associated with each expert's prediction. In the weighted majority algorithm, an expert's weight is updated when it predicts incorrect output. That is,

$$w_i \leftarrow \beta w_i, \text{ if } y_i(x_j) \neq y(x_j),$$

where β is a decaying factor, $y_i(x_j)$ denotes the prediction of i-th expert given input x_j and $y(x_j)$ is the desired output given x_j.

4. Experimental Results and Analysis

The active partitioning method was applied to three data sets: one artificial data set and two real-world data sets from

Table 1: The generalization performance of experts trained with the partitioned sets by RDS and CDS, respectively. (SNN: single neural networks trained with all the training data.)

	Generalization performance				
	Expert				SNN
RDS	57.5	66.3	73.8	82.6	98.3
CDS	96.33	94.50	96.66	95.94	98.3

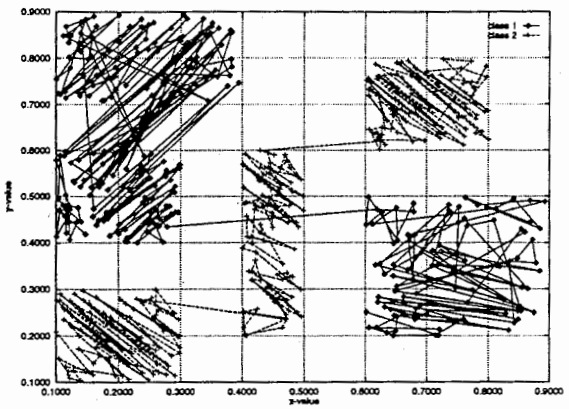


Figure 3: The artificial data and its selection order by RDS. Each point denotes an example. The link indicates the selection sequence. The algorithm partitioned the entire data set into 4 clusters, two for each class.

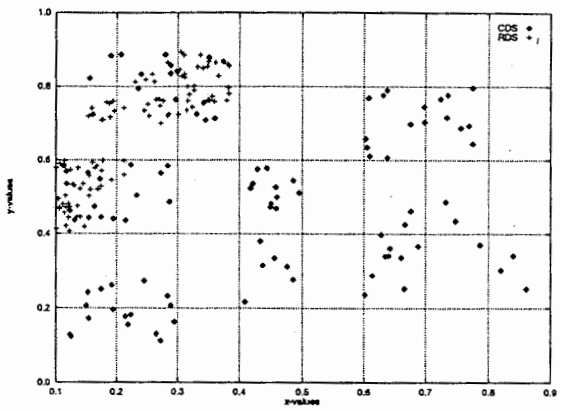


Figure 4: Comparison of the distributions of 100 examples selected by RDS and CDS for the data set shown in Figure 3. It is observed that RDS selects the examples first in one region and then another, while CDS selects examples jumping from region to region.

the UCI machine learning database. The artificial data is used to test whether RDS performs local data partitioning, *i.e.* each subset contains only local information of the problem space. The two real-world data sets are used to analyze the difference of RDS, CDS, and random selection schemes.

4.1. Artificial Data

This is a binary classification problem with two dimensional inputs. As shown in Figure 3, the layout of the patterns looks like the XOR problem but more complex. This problem is originally designed to have 6 clusters, three for each class. When the active partitioning method was applied to this data set, the data was partitioned into two clusters per class.

Figure 3 shows the selection process of RDS. In Figure 3, we can see that RDS selects the data first in one local area, then sequentially moves to another area. Figure 4 compares the distributions of 100 examples selected by RDS and CDS, respectively. While CDS tends to globally select examples, RDS prefers to select local examples. Table 1 compares the generalization performance of the experts which are trained by the subsets partitioned by RDS and CDS, respectively. The CDS experts have better generalization accuracies and their variances are smaller than RDS experts. This result suggests that the experts formed by RDS have the characteristics of local experts and the experts made by CDS have the characteristics of global experts.

4.2. Real-World Data

We also performed experiments on two real-world data sets. One is the australian credit card data set, another is the diabetes data set. The diabetes data set contains 500 examples of class 1 and 268 of class 2. Each example contains eight attributes. The australian credit card assessment problem contains 690 cases in total. The output has two classes. The 14 attributes include six numeric values and eight discrete ones. The reported classification performances of various researches [7] for these two problem are compared with the mixture models. Table 2 summaries the results. The credit card data was partitioned into 15 subsets and the diabetes data into 35. The subpartitioning process is applied to these problems. The number of hidden nodes of each expert is limited to one. Table 3 compares the generalization performances of three methods on the credit card problem. The values are the result of 100 runs for each experiment. RDS did worse than random selection. CDS is competitive to random selection in the mean values but better in the max values. An interesting feature of CDS is that its performance was not affected by the number of experts. This means that CDS represents the distribution of the problem space better than RDS and random selection. Table 4 shows the results for the diabetes problem. In the diabetes problem, the effect is more eminent, and a graphical comparison is given in Figure 5. Table 2 compares the results of CDS with those of other researchers [7] on the two problems. It can be that CDS leads to the best generalization performance except EPNet. The poor performance by RDS seems due to the large variance of generalization performances of experts. The variances for the three partitioning methods are:

$$\sigma^2_{RDS} > \sigma^2_{Random} > \sigma^2_{CDS}$$

We think that the gating network [3] is a better candidate for RDS as a combination method.

Table 2: Comparison of various algorithms in terms of generalization performance on the two problem domains. The performance values for other methods are from Yao [7].

diabetes problem					
Algorithm	CDS	RBF	BP	CART	EPNet
Error Rate	75.60	76.7	74.2	74.5	76.5
credit card problem					
Algorithm	CDS	RBF	BP	CART	EPNet
Error Rate	86.96	85.5	84.6	85.5	88.5

Table 3: Comparison of generalization performance for the credit card problem. The weighted majority algorithm seems to be a bad choice for the experts trained with the RDS partitioning method.

Number of Experts	Redundant Selection (RDS)		
	Mean	Std Dev	Max
3	81.97	4.10	85.67
7	84.63	2.00	87.87
10	84.73	1.51	87.13
13	84.75	0.74	86.11
Number of Experts	Critical Selection (CDS)		
	Mean	Std Dev	Max
3	85.38	0.69	87.28
7	86.36	0.56	87.57
10	86.45	0.43	87.43
13	86.96	0.48	87.72
Number of Experts	Random Selection		
	Mean	Std Dev	Max
3	85.61	0.72	86.99
7	86.13	0.54	87.43
10	86.12	0.49	87.57
13	86.15	0.27	86.70

5. Conclusions and Future Work

This paper presented new data partitioning methods based on active learning. They include CDS which selects the data with maximum error and RDS which selects the data with minimal error. We show that subsets generated by RDS are useful for building local experts and the data subsets produced by CDS are useful for global experts. The experiment shows that the suggested data partitioning methods can be differently used to produce an enhanced generalization performance using a different mixture model. It should be noted that partitioning methods are closely related to combination methods of the experts. The weighted majority algorithm was used in this paper. This algorithm seems useful when the variance of the generalization performances of the experts is not large, as is the case for CDS partitioning. RDS produces local data sets which make local experts. Future work includes finding a good measure for the effective combination of local experts.

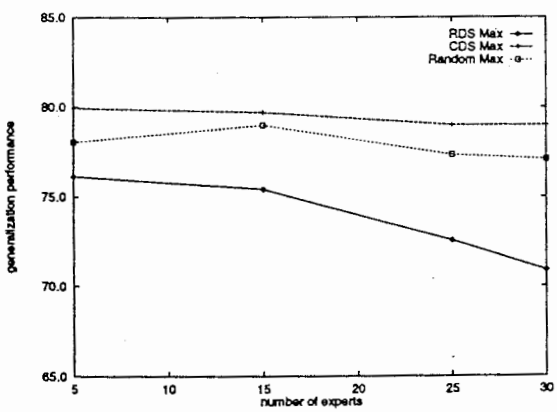


Figure 5: Comparison of max generalization performance for the diabetes problem.

Table 4: Comparison of generalization performances for the diabetes problem.

Number of Experts	Redundant Selection (RDS)		
	Mean	Std Dev	Max
5	69.23	7.19	76.13
15	65.32	6.22	75.42
25	65.36	4.25	72.55
30	65.67	3.51	70.88
Number of Experts	Critical Selection (CDS)		
	Mean	Std Dev	Max
5	71.64	5.72	79.95
15	73.89	2.39	79.71
25	75.11	1.76	79.00
30	75.60	1.39	79.00
Number of Experts	Random Selection		
	Mean	Std Dev	Max
5	72.27	3.23	78.04
15	73.55	2.50	79.00
25	74.03	1.38	77.33
30	74.34	0.81	77.09

Acknowledgement: This research was supported in part by the Korea Foundation of Science and Engineering (KOSEF) under grant 96-0102-13-01-3.

References

[1] Hampshire, J. and Waibel, A., "The meta-pi network: building distributed knowledge representations for robust pattern recognition," Tech. Rep. CMU-CS, pp. 89-166 (1989).

[2] Igor, V. T. and Alessandro, E.P., "Efficient partition of learning data sets for neural network training," *Neural Networks*, **Vol.** 10, No. 8, pp. 1361-1374 (1997).

[3] Jacobs, A.R. and Jordan, M.I., "Adaptive mixtures of local experts," *Neural Computation* **3**, pp. 79-87 (1992).

[4] Plutowski, M. and White, H., "Selecting concise training sets from clean data," *IEEE Transactions on Neural Networks*, **Vol.** 4, pp. 305-318 (1993).

[5] Röbel, A., "The dynamic pattern selection algorithm: effective training and controlled generalization of backpropagation neural networks," Technische Universitat Berlin, Germany, Tech. Rep. (1994).

[6] Shadafan, R.S. and Niranjan, M., "A dynamic neural network architecture by sequential partitioning of the input space," *Neural Computation* **6**, pp. 1202-1222 (1994).

[7] Yao, X. and Liu, Y., "A new evolutionary system for evolving artificial neural networks," *IEEE Trans. on Neural Networks*, **Vol.** 8, No.3, pp. 694-713 (1997).

[8] Zhang, B.T., "Accelerated learning by active example selection," *International Journal of Neural Systems* 5(1):67-75 (1994).

[9] Zhang, B.T., "Convergence and generalization properties of active learning with growing neural nets," *J. Kor. Info. Sci. Soc.* (B), 24(12):1382-1390 (1997).